SimAUD 2015

Washington DC
USA

2015 Proceedings of the
Symposium on Simulation for Architecture and Urban Design

Edited by
Holly Samuelson, Shajay Bhooshan, and Rhys Goldstein

2015 Proceedings of the

Symposium on Simulation for Architecture and Urban Design

Edited by

Holly Samuelson
Shajay Bhooshan
Rhys Goldstein

Cover & Layout by

John Yee

2015 Proceedings of the Symposium for Architecture and Urban Design
Holly Samuelson, Shajay Bhooshan, and Rhys Goldstein, editors

ISBN: 978-1-329-04938-3

Contents

Simulating Human Behavior in not-yet Built Environments by means of Event-based Narratives

Davide Schaumann[1], Yehuda E. Kalay[1], Seung Wan Hong[2], and Davide Simeone[3]

[1]Technion – Israel Institute of Technology, Haifa, Israel
{deiv, kalay}@tx.technion.ac.il

[2]Inha University
Incheon, South Korea
seungwanhong75@gmail.com

[3]Sapienza University of Rome
Rome, Italy
davide.simeone@uniroma1.it

ABSTRACT
Current Computer-Aided Architectural Design (CAAD) systems fail to represent buildings in-use before their realization. This failure prevents testing the extent to which a proposed setting supports the activities of its intended users. We present a novel approach to human behavior simulation based on a thorough representation of end-user activities by means of *events* – computational constructs that simulate users' individual and group activities to achieve a specific goal. Human behavior narratives result from a combination of *top-down* (planned) and *bottom-up* (unplanned) sequences of events, as a reaction to time-based schedules and to social and environmental stimuli, respectively. A narrative management system orchestrates the narrative developments and resolves conflicts that may arise among competing events.

Author Keywords
Event-based model; Virtual users; Human behavior simulation; Building design evaluation.

ACM Classification Keywords
I.2.11. [Artificial Intelligence]: Distributed Artificial Intelligence; I.6.5 [Simulation and Modeling]: Model Development; I.6.8 [Simulation and Modeling]: Types of Simulation; J.6 [Computer-Aided Engineering]

INTRODUCTION
During the design process architects and clients need to determine in what ways a building will support end-user activities; a building that does not meet its user needs will suffer in terms of lack of functionality, waste of space, delays in task accomplishments, and user dissatisfaction. Nevertheless, predicting and evaluating such building performance is a complex task: user behavior is driven by work-related tasks within an organization, by social and environmental aspects related to the surrounding context, and by individual desires and motives, mediated by perceptual, cognitive as well as cultural and social factors.

Current approaches to human behavior simulation mainly rely on the Agent-Based Modeling (ABM) paradigm, which focuses on assessing specific responses of human inhabitants to social and environmental stimuli, such as fire egress situations [8] and pedestrian movement [5]. Nonetheless, ABM has shown conceptual and technological limits in representing more comprehensive buildings' spatial use patterns, involving both scheduled activities and more serendipitous one (e.g. social interactions), despite the fact that both types of activities are the bulk of everyday functioning of a building.

To overcome such limitations, Simeone & Kalay [17] proposed a simulation model based on the notion of *event*: a 3-tuple computational structure that combines *actors*, *activities*, and *spaces* into a single, holistic unit. This model proved its capabilities in simulating *top-down* pre-planned logical sequences of events. We aim at expanding the capabilities of the model to afford intertwining *top-down* scheduled sequences of events with a *bottom-up* lists of events not scheduled ahead. Bottom-up events are performed in relation to agents' individual traits (e.g. an agent's health condition) or to groups' serendipitous situations (e.g. social encounters). These aspects are of critical importance especially for representing human behavior in complex, social environments such as hospitals. Within these settings activities due to social and contextual occurrences may have serious consequences for the function of the environment, and may result in diversions from usual work processes [14].

In this paper we elaborate on the notion of events as computational behavior units, and we propose a narrative management system to simulate both planned and unplanned activities in hospital environments.

STATE OF THE ART

Human behavior in built environments
Architects' domain of expertise lies within the realm of configuration and prefiguration of physical settings. The built environment, in turn, considerably influences the behavior of its users. Several research practices investigated the relationship between people and their physical environment, including the field of Environmental Psychology. Barker, one of its founders, proposed a comprehensive approach to investigate the dynamic interplay between people's goal-oriented activities and the environment in which they are performed [3]. The approach relies on the notion of *behavior setting*: a bounded, self-regulated system that involves both human and non-human components synchronously interacting to carry out *behavior*

SimAUD 2015, April 12 - 15, 2015, Alexandria, VA, USA

units, defined as temporal and logical sequences of events [21]. Behavior units are aggregated in observable standing patterns of behavior dependent on the social "program" of the setting rather than on people's individual decision-making processes. Furthermore, these behavior patterns represent a consistent activity system that adapts to changes in the composition and number of participants, or in relation to the spatial features of the physical setting.

Following Barker's theory of behavior setting and Alexander's pattern-based approach [1], we argue that standing patterns of behavior commensurate with environmental, social, and cultural contexts can be computationally represented along with the physical settings in which they occur. These patterns, after being observed in real environments and computationally formalized, can be used to test by means of simulation the affordance of a setting to support or hinder the performing of the related activities.

Simulating behavior narratives

Human behavior narratives can be defined as logical, time-based activity structures evolving in time in a coherent fashion. In the architectural field, current methods to simulate human behavior narratives involve two opposite approaches: the first consists in a *top-down* deterministic definition of time-based schedules to simulate building spatial utilization processes [18], buildings' occupant behavior in relation to energy use [10, 12], and designer-user communication [15]. The second consists in a *bottom-up* approach to provide agents with simple rules of behavior in response to environmental and perceptual stimuli [19].

In the video game industry, instead, much effort is dedicated to mix the two aforementioned approaches by creating intelligent Non-Player Characters (NPC) that follow a general narrative script, and that are capable of dynamically adapting their actions in reaction to a player choices. Central narrative management systems orchestrate a narrative development according to agent goals, and environmental conditions and properties [4, 13]. However, drawbacks involve the complexity of encoding narrative management rules, and high computational costs. Alternatively, hybrid narrative management systems distribute decision-making tasks between a narrative manager and semi-intelligent characters [7, 20, 9]. This approach advocates a division between a top-down *story-level* control system and a bottom-up *character-level* system. The integration between these two levels can occur by means of small-granularity story units, which represent individual and group behaviors [2, 11, 16].

In this paper we elaborate on a model proposed by Simeone & Kalay [17], which adopted video game hybrid narrative management strategies to simulate human behavior narratives in not-yet built environments. The model, designed to overcome the limitations of previous ABM approaches, trades some aspects of single agents' decision-making abilities in favor of a centralized direction mechanism, affording authorial control over the performing of complex operations such as group activities and agent collaboration. The model relies on the notion of *event*—a computational structure that embeds the knowledge required to perform individual and collaborative tasks in virtual settings. When triggered, an event temporarily reduces the autonomy of the involved agents to direct them through a series of actions in a coordinated manner.

This narrative-oriented interpretation of the word "event" is consistent with previous work in the video game industry [16]. A somewhat different interpretation is found in the discrete-event simulation literature, where an "event" is still a representation but occurs at an instant of time. However the self-contained nature of an event is common to both discrete-event simulation and the narrative-oriented context of our work. Furthermore, the two primary event-triggering mechanisms in discrete-event simulation—the scheduling of events and the receiving of new information—are analogous to the planned and unplanned events described in this paper.

EVENTS AS BEHAVIOR UNITS

Events combine three types of information: the *actors* that populate a setting, the *activity* they do, and the *space* they use. By juxtaposing entities among these non-homogeneous information domains events define behavior units with context-related semantics, and describe segments of the use processes that occur in buildings in relation to their function (Figure 1). For instance, an event describing a common activity within a hospital setting could be: "visitors" (*actors*) "talking to patient" (*activity*) in "patient room" (*space*).

Figure 1. Event coordination mechanism

A distributed intelligence approach

To reduce events' computational efforts in managing the performing of a behavior pattern, each of the aforementioned event constituents, namely the *actors*, *spaces* and *activities*, presents dynamic calculation capabilities. The event—acting like an "orchestra director" —provides top-down management of the aforementioned involved entities to assure the achievement of the goal.

Space

To support human behavior simulation, we propose to augment the spatial representations afforded by current Computer Aided Design (CAD) and Building Information Modeling (BIM) tools by adding semantic and environmental information. Such information is continuously updated in accordance with the activities and environmental conditions prevailing in each space at a given time.

Recent attempts to encode space semantics in a static fashion (e.g. through IFC models) have been proven inadequate for the dynamic simulation of user activities within predefined space boundaries (e.g. a room, or a specific area within a room). Rather, a time-dependent representation is required, since a specific space may assume different meanings depending on the nature of the activity that occurs in it, or the composition of the actors performing that activity. For instance, if an event called "patient check," which involves a doctor and a nurse checking a patient, is performed in a hospital room, the character of the space becomes akin to a clinic, which excludes visitors. Once the "patient check" activity is over, visitors may be allowed, since the character of the space changes into "patient room."

A pre-coded list of possible semantics defines each space affordance in terms of supporting a discrete number of activities. During the simulation process, space entities are responsible for detecting the activity performed within their boundaries, and selecting the corresponding semantic value, therefore updating their own properties. Conflicts among competing semantic values attributed to the same space in relation to multiple activities are solved by means of a rules-based system, which determines the current space semantics in relation to multiple parameters such as event priorities, as well as social and environmental factors.

Environmental data is added in the form of *data-maps*, dynamic databases that represent environmental properties such as noise, light, smell, and density of people. Data-maps are updated at specific time intervals during the simulation process in response to environmental conditions, and are queried by events to determine whether the conditions required to perform a specific behavior are fulfilled. Changes in both space semantics and data-maps values can, in fact, cause an event to adopt a different strategy to achieve its goal, or even be cancelled if an alternate plan is not available.

Actors

Anthropomorphic goal-oriented virtual users mimic end-user behavior in a virtual setting [19]. Each agent, called *actor*, embeds geometrical properties, a semantic role he/she occupies in the organization (e.g. doctor, or nurse), and physiological and psychological traits whose values remain fixed during the simulation process (e.g. age, gender, experience) or vary dynamically according to the activities performed (e.g. tiredness, stress, hunger). As with spatial semantic information and data-maps, event entities access information stored in actor profiles to trigger or adapt the performing of an activity in relation to individual features.

Actors are capable of *low-level* decision-making to perform activities such as path finding and walking towards a target. For instance, to move through a virtual setting agents rely on their physical and psychological condition, as well as on their role in the organization (e.g. a doctor, or a nurse), which affects their environmental knowledge.

Environmental perception and cognition are not encoded in actors themselves, but rather are mediated through a spatial "awareness" system that detects actor and object presence in spatial zones. For example, to determine whether two agents are present in the same room, an event communicates with the "room" entity itself: the room has the capacity to detect which agents are located within its boundary, in a manner that is simpler than endowing each agent with spatial awareness capabilities. This approach ignores sight lines and gaze direction, as is done in other research projects at a much higher computational cost [6]. We consider this approach an acceptable trade-off for the purposes of our simulation.

Basic constructs involving actor groups are managed by event entities, which can for instance coordinate the behaviors of patients' family members while visiting their relative. In this case, the semantic role of the group members and the relationship between them play an active role in performing the task.

Activities

Activities provide a set of actions and procedures that direct agents toward the accomplishment of individual or group tasks. They consist of methods/functions that are called by events, and that take as arguments at least an *actor* entity and a *space* entity associated with the same event. Parametric values guide the performing of activities, and allow them to be reused in different circumstances by multiple actors. For example, the activity "talk to" takes as arguments the *actors* involved (at least two), the *semantics* of the space in which the activity can be performed, and the *duration* of the action. Performing activities that involve an agent movement through space, such as path finding, obstacle avoidance, and speed of motion, rely on each agent individual capabilities. In case of activities involving a group of actors moving towards a target, the event entity coordinating their behavior selects a "group leader" who will share with the other group members his/her path-finding capabilities.

Event performing structure

Events are implemented as self-contained autonomous routines with decision-making capabilities in relation to the status of the entities involved, the social and physical context, and stochastic processes. They include *pre-conditions*, a set *of performing procedures*, and *post-conditions*.

Pre-conditions specify the requirements for an event to be triggered. They might be related to the activity, space, or actor entities involved in the event, or to more general constructs, such as *time*. Preconditions concerning space, for instance, might check for the space semantics, data-maps values, or other information detected by the spatial awareness mechanism.

After verifying the compliance with the preconditions, a set of *performing procedures* guides the event execution. Such procedures are provided by an event's *activity* component. An activity success test notifies the event about a task achievement when terminating conditions are satisfied.

Upon termination, events update the status of all the entities involved in its execution (such as actors, spaces, furniture) by means of *post-condition* instructions.

In case the preconditions of a sub-event are not met, the event seeks for an alternative plan stored inside its performing procedure to achieve the same task. If such plan cannot be found the event is aborted.

EVENT-BASED NARRATIVES

By combining events into larger compositions we generate human behavior narratives related to how buildings are used by their occupants. Narratives provide a logical plot structure that unfolds during the simulation process according to event preconditions, as well as to stochastic processes. To define narratives we use a system of nested events, assembled in logical structures.

Event nesting

Events can be nested within other events in a tree-like manner (Figure 2). This approach aims to increase the level of detail in representing behavior units, and to make the complexity of a building use-process more manageable. The nesting system allows defining behaviors as a composition of sub-behaviors whose performing eventually leads to a task achievement. A parent event is responsible for orchestrating the performing of its children events so that sub-events do not need to be aware of their siblings. In this manner, parent events are able to make informed decisions about the performing of their sub-events. This system affords a hierarchical control mechanism in which rules and properties encoded in a parent event propagate to the children. The nesting system allows representation of behaviors at different levels of abstraction for the purpose of managing increasing complexity and adapting to the level of detail required by the simulation to describe human behavior phenomena.

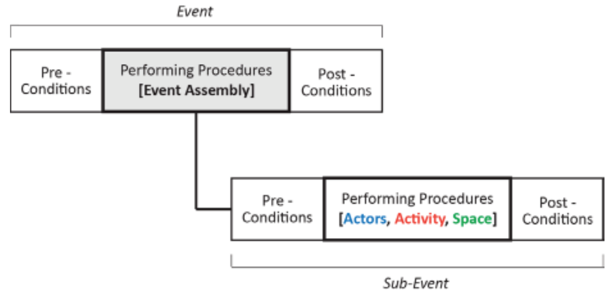

Figure 2. Event nesting

Event assemblies

Multiple sub-events are nested within the performing procedures of a parent event. We define this composition as *event assembly*. Event assemblies define the execution logic of a structured set of events to achieve a task stated by the parent event. Within the assemblies, events are combined by means of logical operators, which are embedded in the parent event, and describe the rules to perform the children events. Logical operators are of three kinds: *sequence*, *parallel*, and *selection* (Figure 3).

The *sequence* operator defines a logical and temporal sequence among events that occur one after the other.

The *parallel* operator assembles two or more events that are triggered at the same time. This type of operator is useful to synchronize two activities starting at the same time.

The *selection* operator indicates a choice that has to be made at the parent node about which sub-events to trigger next. The selection is realized by evaluating the event preconditions and priority values. A stochastic mechanism can also be applied to randomize the selection process. In case of an event's failure to complete a task, an alternative plan to achieve the same task can be stored within the parent node, and selected by means of this operator type.

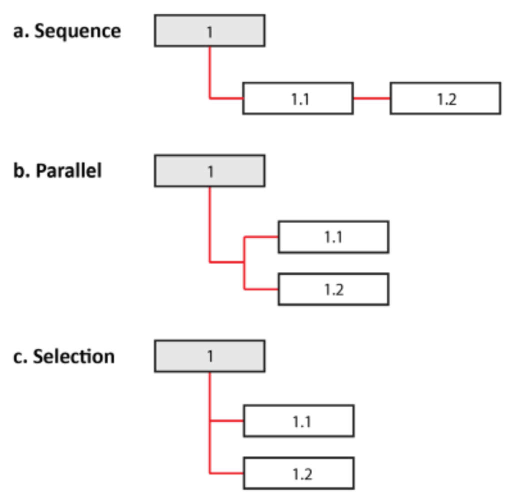

Figure 3. Logical operators to define event assemblies

Example: "Patient Check" event

To illustrate the concepts of event *performing*, *nesting* and *assembling*, we discuss the "patient check" event (Figure 4), which is a common event in a general hospital ward. "Patient Check" is considered the parent node of a nested event assembly comprising a sequence of two events (1.1 & 1.2). Event 1.1 is in turn the parent of two sub-events (1.1.1 & 1.1.2), combined by a parallel operator. These events direct a doctor and a nurse (*actors*) through a specific *space* type (a corridor), towards the room of the patient (another *actor*) who needs to be checked (*activity*). The space type defines the required semantics to support the activity performing. Sudden emergency situations taking place in the corridor might in fact change the semantics of the space through which the doctor and the nurse are moving, obliging the actors to look for an alternate route to reach their target (or abort the "patient check" event in favor of engaging in an emergency event). After completing Events 1.1.1 & 1.1.2, the "move to patient bed" event (1.1) is fulfilled, and the narrative proceeds to Event 1.2, which comprises two nested events (1.2.1 & 1.2.2) connected by a selection operator. After evaluating both event preconditions, Event 1.2 decides which one of the two sub-events to trigger. Event 1.2.1 involves both doctor and nurse checking on a patient in a space whose semantic value is "clinic". As mentioned before, any changes in the space semantics, such as a visitor talking loudly in the same room can prevent the patient check event from being performed. If the conditions to perform Event 1.2.1 are not satisfied (e.g. the patient is not in the bed), the event will be aborted. After navigating the event graph till its end, the control is brought back to the root event (Event 1). The root event updates a loop counter stored within its performing procedures indicating the patients that have already been checked, and the ones that yet have to be checked. Event 1 therefore re-triggers the same sub-events for the next patient to be checked, until all patients have been visited.

NARRATIVE MANAGEMENT

A hierarchical event composition structure is used to simulate *use-process* narratives in buildings (Figure 5). The system consists of a distributed decision-making mechanism that involves *actors* at a lower level (*operational)*, *events* at a middle level (*tactical)*, and a *narrative manager* at the higher level (*strategic)*. At the *operational* level, behavior is directed by actors' individual traits, such as walking speed, and path-finding capabilities. At the *tactical* level, events coordinate actors' behavior to accomplish a task. At the *strategic* level, a narrative management system directs the evolution of the simulation narrative in a coherent fashion by determining which event to perform next. For management purposes, events are classified according to whether they are planned or they are selected due to impromptu circumstances, and whether they are performed by individuals or by groups of actors.

Planned events are scheduled in time (e.g. a meeting), and they represent the procedures that designers aim at simulating and evaluating. *Unplanned events* are triggered in relation to contingent situations (e.g. users' physiological needs, social interactions, or emergency situations), and they lead to deviations from the performing of planned events. *Individual events* involve a single agent and are performed without interacting with others, whereas *group events* involve several agents with a shared common goal.

This system organization allows distributing decision rules at different levels of the narrative to resolve conflicts that arise among competing events. Higher nodes in the event tree solve conflicts among events in the lower hierarchy level by means of a rule-based system that indicates which event to perform among two competing ones.

Planned events

Planned events are encoded in the form of a top-down comprehensive time-based schedule. Events starting at the same time are nested within a higher-level event that triggers all the sub-events by means of a parallel operator, when the time preconditions are satisfied. When an event time preconditions are satisfied, a daemon triggers the respective event(s).

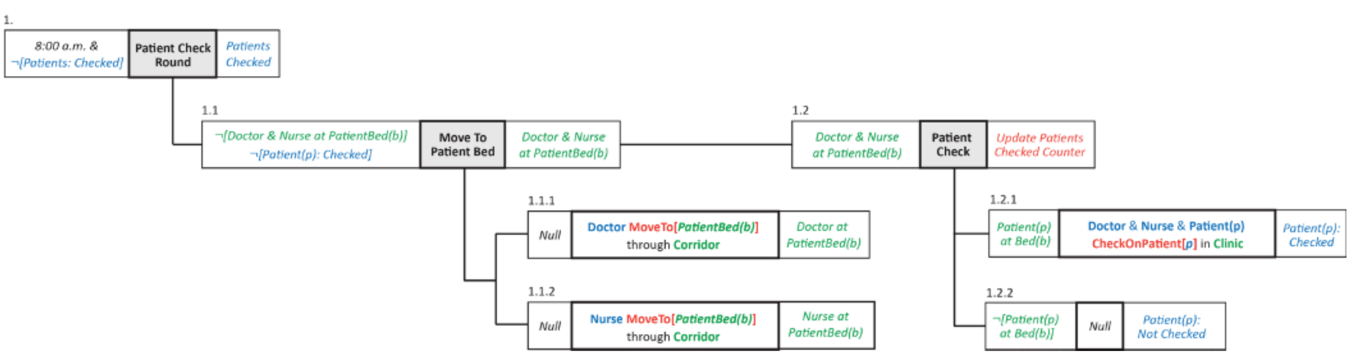

Figure 4: Representation of a "patient check" event. The diagram demonstrates the event nesting and assembly mechanism. Each color identifies a different domain where information is stored or calculated: information on *actors* is represented in blue, on *spaces* in green, and on *activities* in red.

The higher-level event (Event 1.1 in Figure 5) combines the information encoded in the children nodes to generate a comprehensive time-based schedule that accounts for both individual and group behaviors. Furthermore, Event 1.1 consolidates decision-making capabilities involving both group and individual schedules, such as rescheduling an event due to conflicts or delays.

Unplanned events

Unplanned events are encoded in the form of a list of possible behaviors that may be performed when determined preconditions (involving spatial, environmental, or social stimuli) are satisfied.

Unplanned group events (1.2.1) stipulate that actors must collaborate in order to achieve a joint task. *Social interaction events* (1.2.1.1), for example, can be triggered when two actors are within a predefined proximity of one another. To verify this condition, a daemon constantly monitors a *proximity data-map*, which detects the position of actors in space. When the conditions are fulfilled, the daemon will notify the event.

Instead, a list of unplanned individual events is defined for each actor, and nested within a parent event (1.2.2.1 & 1.2.2.2). Such events can be triggered in two ways: the first when a daemon notices some conditions, and requests other events' interruption. For instance, a medical emergency (so-called "Code Blue") event is triggered when a daemon that monitors the vital signs of each patient detects a health-threatening situation (Event 1.2.2.2.1). This event involves directing a team of doctors towards the patient to initiate resuscitation procedures. The second way is when an actor is not involved in any other planned event. In this case, to perform a selection, the parent event, which resembles individual actors decision-making capabilities, checks the children-events' preconditions accounting for the individual agent's drives and motives (e.g. doing his job, assisting others etc.). In case where two or more events are suitable for performing, the parent event will evaluate the events' priority values.

Narrative Manager

A *narrative manager* oversees the performing of all the events, and manages information that is shared among different levels of the events hierarchy, such as the time parameter and daemons' actions. Due to its root position in the narrative hierarchy, the narrative manager resolves conflicts that arise between competing planned and unplanned events. To resolve a conflict, a rule database, which is stored within the narrative manager's performing procedures, is consulted. Rules define which event to perform among competing events, and under which conditions. For instance, if the conditions to perform a social interaction event are satisfied while an agent is already performing another event, the narrative manager detects a conflict. Its resolution depends on many parameters, including event priorities and the actor individual states (e.g. stress, tiredness). The evaluation of these parameters will determine if a social event can be triggered (e.g. the doctors will pause to talk to each other), and, if so, for how long.

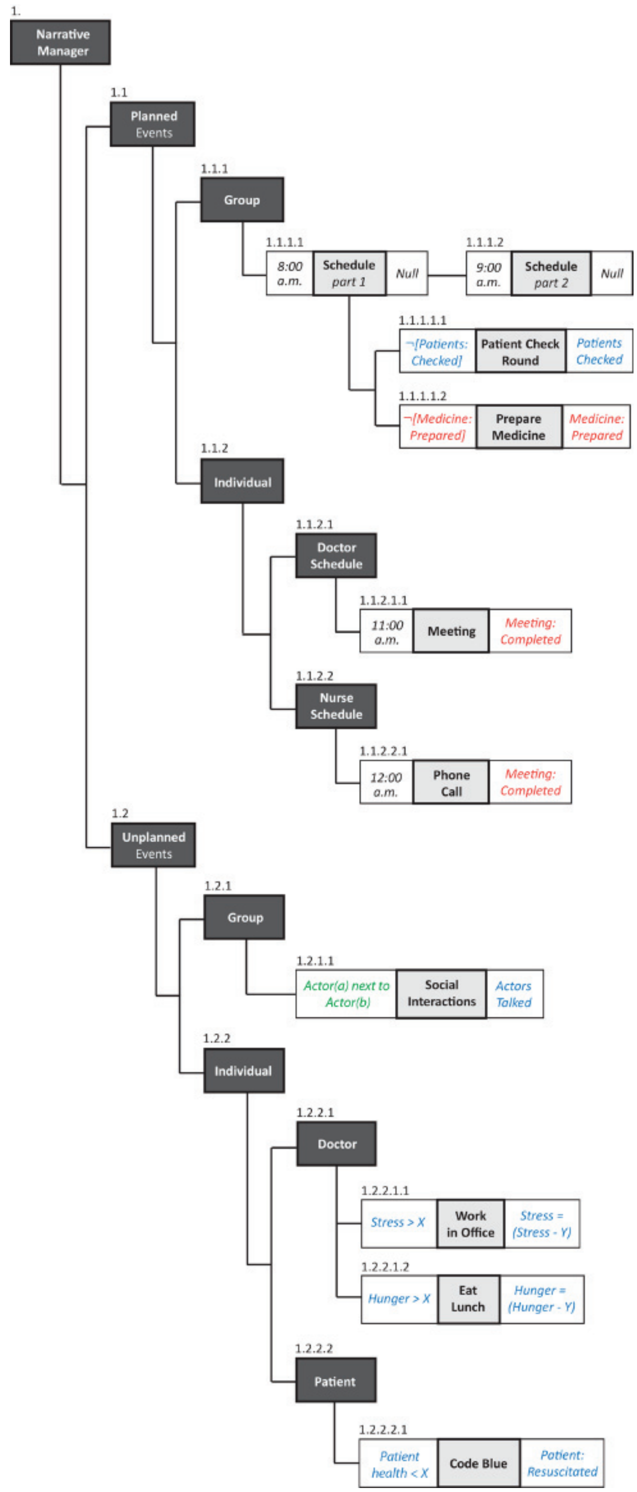

Figure 5: Narrative management system defined by Planned & Unplanned events, and by Group & Individual events.

This conflict resolution mechanism aims at obviating the cumbersome procedure of dynamically adapting event priorities during the simulation process in relation to the status of the actors involved and of the environment in which they operate. While event priorities remain constant, more rules can be added that relate to different aspects of the environment, to make more informed decision in resolving conflicts.

CASE STUDY

The following case study is offered to elucidate the simulation model described in this paper (Figure 6). The simulation involves four actors in their daily routines. The setting consists in a simplified hospital environment where activities typically adhere to strict time-based schedules, interrupted by unexpected events. The spatial layout comprises several semantically different zones. No planned activities are scheduled at the beginning of the simulation. Rather, each actor autonomously selects which event to perform (Figure 6a). At the appointed time, the planned event "patient check" is activated. It directs a doctor and a nurse to go to the patient room, where a patient is talking to a visitor. The arrival of the doctor and the nurse at the patient room causes a change in the room semantics, from being a *patient room*, to a *clinic*. This change causes the event "patient talking to visitor" to stop, and the visitor to leave the room. When the doctor and nurse are close to the patient bed and the related preconditions are satisfied, the "patient check" event takes control of the three actors to perform the collaborative activity (Figure 6b). After completing the event, each actor goes back to resume his/her previous task. On the way to his office, though, the doctor passes close to the visitor, who is waiting in the lobby. The spatial awareness system of the space zone "i" detects the proximity of the two agents and communicates it to a "social interaction" event, which tests additional preconditions concerning the doctor's state of stress, and the visitor's anxiousness value. Since the conditions are verified, the event is performed (Figure 6c).

This simulation was facilitated by a host of computational tools: Autodesk Revit allowed the physical setting's geometrical data modeling; Autodesk 3DS Max allowed actors' geometrical data modeling and animation generation; Unity 3D – a video game engine tool - provided a simulation environment with dynamic visualization capabilities. Data-maps, semantics, agent profiles, activities, and events were also scripted in Unity.

CONCLUSIONS AND FUTURE DEVELOPMENTS

During the design process, architects rely on their own, partial and biased experience to foresee the impact of a physical setting on future human behavior. The research presented in this paper outlines a model to simulate human behavior in buildings before construction, for evaluation purposes. The model relies on *events*—self-contained behavior units that are assembled into larger narratives accounting both for *top-down* (planned) and *bottom-up* (unplanned) activities. A hierarchical control system controls the progression of human behavior narratives, and resolves conflicts between planned and unplanned events. The simulation accounts for the effect produced on the activity performing due to changes in environmental properties and semantic meaning. We argue that the model allows for a better management of human simulation narratives, guaranteeing flexibility in behavior design, and allowing increasing level of detail.

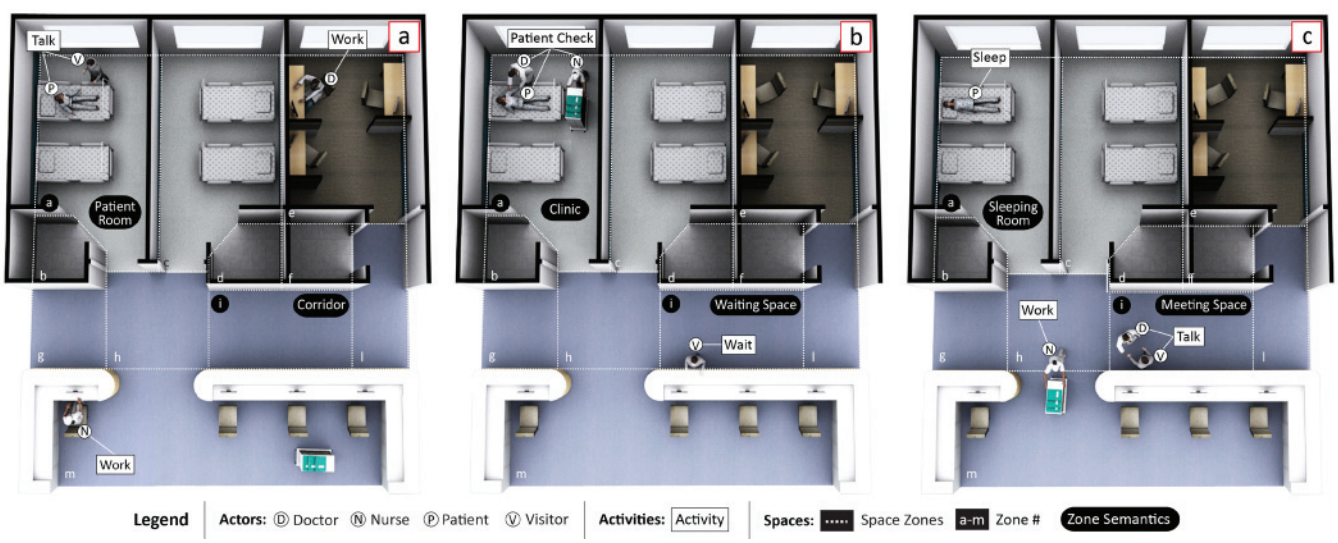

Figure 6: Simulation of a "patient check" event. The sequence describes the narrative management system, which dynamically interweaves planned and unplanned events, and responds to changes in space semantics.

Further work is required at all levels of the model: at the *agent* level, the enhancement of visibility calculations; at the *event* level, the development of better strategies to evaluate which event to perform among a list of options in relation to probability considerations and utility functions; at the *narrative* level, the definition of the appropriate size of event assemblies to guarantee an optimal management system.

Future developments will involve the creation of a more comprehensive system to include other state-of-the-art simulation methodologies, and the application of such system to simulate other facility types, such as schools, airports, or museums. More generally, we look forward to develop a computational platform test the impact produced by a built environment on its inhabitants, implementing human behavior knowledge deriving from multiple research fields, such as Cognitive Science, Social Science, and Environmental Psychology.

ACKNOWLEDGMENTS

This research is kindly supported by a European Research Council grant (FP-7 ADG 908040). We are also grateful to the following research group members for their useful comments and insights: M. Brodeschi, M. & E. Morad, N. Pilosof, B. Plotnikov, and H. Soffer.

REFERENCES

1. Alexander, C. *The Timeless Way of Building*, Oxford University Press, 1979.
2. Badler, N., Bindiganavale, R., Allbeck, J., Schuler, W., Zhao, L., Palmer, M., A parameterized action representation for virtual human agents. In: *Embodied Conversational Agents*, (eds.) Cassell, J., MIT Press, Cambridge, MA, 2000, 256–284
3. Barker, R. G., *Ecological Psychology: concepts and methods for studying the environment of human behavior*. Stanford University Press, 1968.
4. Bates, J., Virtual reality, art, and entertainment. *Presence: The Journal of Tele-operators and Virtual Environments 1,1* (1992).
5. Batty, M., Agent-based pedestrian modeling. *Environment and Planning B: Planning and Design*, *28*, 3 (2001), 321-326.
6. Bhatt, M., Schultz, C., & Huang, M. The shape of empty space: Human-centred cognitive foundations in computing for spatial design. In *IEEE*, *Visual Languages and Human-Centric Computing (VL/HCC), 2012, 33-40.*
7. Blumberg, B. M., and Galyean, T. A., Multi-level direction of autonomous creatures for real-time virtual environments. *Proc. Computer graphics and interactive techniques*, ACM Press, 1995.
8. Chu, M. L., Parigi, P., Latombe, J. and Law, K.. SAFEgress: A Flexible Platform to Study the Effect of Human and Social Behaviors on Egress Performance. *Proc. SimAUD*, (2014), 35-42.
9. Magerko, B., Laird, J., Assanie, M., Kerfoot, A., and Stokes, D., AI characters and directors for interactive computer games. *Ann Arbor* 1001,48 (2004), 109-211.
10. Mahdavi, A., Simulation-based control of building systems operation. *Building and Environment 36*, 6 (2001), 789-796.
11. Mateas, M., and Stern, A., A behavior language for story-based believable agents. *IEEE Intelligent Systems* 17, 4 (2002), 39-47.
12. Reinhart, C. F., Lightswitch-2002: a model for manual and automated control of electric lighting and blinds. *Solar Energy* 77, 1 (2004), 15-28.
13. Riedl, M.O., Stern, A., Dini, D. M., Alderman, J. M., and Rey, M. D., Dynamic Experience Management in Virtual Worlds for Entertainment, Education, and Training. *Int. Transactions on Systems Science and Applications 3*, 1 (2008), 23–42.
14. Seo, H. B., Choi, Y. S., & Zimring, C. (2010). Impact of hospital unit design for patient-centered care on nurses' behavior. *Environment and Behavior 43,4* (2011), 443-468.
15. Shen, W., Shen, Q., & Sun, Q., Building Information Modeling-based user activity simulation and evaluation method for improving designer–user communications. *Automation in Construction 21*, (2012), 148-160.
16. Shoulson, A., Francisco M. Garcia, F.M. Jones, M., Mead, R., and Badler, N.I. Parameterizing behavior trees. *Motion in Games*. Springer Berlin Heidelberg, (2011), 144-155.
17. Simeone, D. and Kalay, Y.E., An Event-Based Model to simulate human behaviour in built environments. *Proc. eCAADe,* (2012), 525-532.
18. Tabak, V., de Vries, B. and Dijkstra, J., Simulation and validation of human movement in building spaces. *Environment and planning. B, Planning & design 37,4* (2010), 592-609.
19. Yan, W. and Kalay, Y. E., Simulating Human Behavior in Built Environments. *CAAD Futures,* (2005), 301-310.
20. Young, R. M., An overview of the mimesis architecture: Integrating intelligent narrative control into an existing gaming environment. *The Working AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment* (2001), 78-81.
21. Wicker, A. Ecological psychology: historical contexts, current conception, prospective directions. In Bechtel and Churchman (Eds). *Handbook of Environmental Psychology*, NY: John Wiley and Sons, 2002.